# Introduction

Thanks to the increase in storage capacity, the decrease in price and the rapid development of computing power, deep neural networks (DNNs) have been used on a large scale. Follower deep learning networks are used in more and more sensitive fields, such as face recognition【】, object recognition【】, and semantic segmentation【】.

with convolutional neural networks are used on a large scale in production, networks will be deployed in many domains, which are different from training dataset, such as distribution. Due to the huge cost of network training, models are required to be able to be applied in domain which are not encountered when training. A survey of domain generalizations proposes a clear definition of domain generalization, with the goal of learning a model from a given several different but related domains, which exhibits good performance on unseen domains. There are many methods can achieve this goal: the simplest one is neglecting the trial of domains[erm]; [mixup的作者] propose create a virtual sample to improve the ability of generalization;[rcs 作者] ;[vrex 作者]

Because the training of network requires a lot of computing power to achieve good results, which cause the training tasks are often done by a third party. In this case, the third party can change the behavior of the model by poisoning the training dataset invisible. By injecting a small amount of poisoned data, the model can be made to make bad decisions in certain situations without changing its performance on a clean test set.

Or if user trains the model and collects data themselves, that means the training dataset may be collected in many ways which can be attacked in malicious. The model may have same performance in certain situations. Therefore, the domain generalization will be attacked easier because of its multiple domains.

Because of the trial of DG, will the backdoor can be implanted into model when just some domains are attacked, or only all domains are attacked, the backdoor can be implanted. And because of different methods used in domain generalization, will these methods affect the performance of backdoor implanted in model, or even improve the backdoor’s performance. What’s more, when domains are attacked and model has been trained by using these domains, can we use the model to distinguish attack samples and can we remove the backdoor from model. To solve these problems, we first attack different numbers’ domain while training model with different method[mixup erm rcs vrex], then using test dataset with attack sample to assess whether the backdoor is implanted into model. The experiment shows numbers of attack domain don’t affect the backdoor implant, and different domain generalization methods also have no influent on performance of backdoor.

Base on the result above, we propose using activity cluster[ac] to clear out domains which are attacked and which aren’t, and using domains without attacking to help us distinguish attack samples. Experiment show that we can point out domains with attack and find out most attack samples by using our method, that is the backdoor can be remove from the model.

Therefore, our contributions are in blow:

* We prove backdoor can be implanted into model just by attacking some domains or even one domain, and the method for domain generalization has no effect on the backdoor’s implanting and performance.
* We propose to use domains without attack to guide how to distinguish datasets with mixed attack and clean sample.
* Through experiments and evaluations on multiple datasets, we demonstrate that using clean datasets to guide the culling of poisoned data is highly accurate and efficient.

# Related work

In this section, we will introduce domain generalization, then we will describe our threat model and detection algorithm in belief.

Domain generalization

Domain generalization is proposed for training in different domains, which have different distributions, to gain a model which can be used in other domains and perform well. In recent years, it has received a lot of attention in deep learning. A simple domain generalization method is to construct a new but fictitious sample [mixup] by mixing data from different domains to train a model with better performance when it’s deployed unseen domains; The distribution between the samples of different domains are different[mmd], which makes the model perform poorly when facing the data of the unmet domain, so the proposal to maximize the difference in mean can make the model show better performance in the face of unknown domains. In addition, a pruning-like process is used to allow the model to learn more content-related features to eliminate the influence of different domain environments on model judgment [rsc].

Threat model

A backdoor attack is usually thought of as the trainer use dataset with attack samples which consists of clean samples and triggers to train a malicious model ( ) that will correctly classify clean samples; but for any attack samples, the model will cause it to be misclassified as target class, or directly reduce the classification accuracy. Backdoor attack usually achieves by attacking samples. The simplest method is injecting some signs into samples, and this is usually implemented by adding a signal bright pixel to samples[badnet] or using an encoder to produce a trigger which depends on sample’s trial and mixing it with sample to produce attack samples[reflect].

Backdoor defense

Backdoor defense is usually thought of eliminating the backdoor implanted in the model, to help model classify all samples in right. Meanwhile, the removal of backdoor should not decline the accuracy of classifying clean samples, or the decline can be ignored. The usual methods used in removal are distinguishing the attack samples by their characters which different clean samples, like using the spectral signature[neruips 2018]; another methods are dropping out the attacked neuron to remove the backdoor[19 20 21].

# Preliminary

To formally beginning to solve our question on whether and how can we distinguish attack samples and clean samples by using clean domains, we should do some experiment to answer what the attack domains’ number and method of domain generalization will affect the procession of backdoor implant and its performance.

The experiment’s setting is ResNet-12, proportion of attack samples is 10%, PACS is used as our dataset and we let all attack samples are misclassified to target class(this paper target is class 0). We try different methods, and each domain generalization method will attack 1~d-1 domains respectively (where d is the total number of domains in the dataset). The results [Table 1 ] show that different numbers of attack domains do not seem to affect the implantation of the backdoor, and the backdoor can be successfully implanted into the model even if only one domain is attacked. The table also show the backdoor will perform well when meeting clean samples or attack samples. More importantly, the methods of domain generalization have no influent on backdoor implantation.

Table 1: We record the results of the experiment which are based on above setting. The attack dispersion indicated the number of attacked domains, the test acc and test acc(poisoned) mean the classified accuracy of clean samples and attacked samples respectively. The high latter value means performance of the backdoor is bad. The acc drop can be calculate by using formulate blow: (test acc- test acc(poisoned))/test acc, this is another indicator show the performance of backdoor.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| algorithm | Attack rate | Attack dispersion | Test acc | Test acc(poisoned) | Acc drop (%) |
| ERM | 0 | / | 0.8003 | 0.7998 | 0.06 |
| 0.1 | 1 | 0.7988 | 0.2358 | 70.48 |
| 0.1 | 2 | 0.7925 | 0.1968 | 75.17 |
| 0.1 | 3 | 0.8027 | 0.2012 | 74.93 |
| RCS | 0 | / | 0.8022 | 0.8018 | 0.05 |
| 0.1 | 1 | 0.7848 | 0.2073 | 73.59 |
| 0.1 | 2 | 0.7612 | 0.1897 | 75.08 |
| 0.1 | 3 | 0.7705 | 0.1943 | 74.78 |
| MMD | 0 | / | 0.7979 | 0.7993 | -0.18 |
| 0.1 | 1 | 0.7798 | 0.1940 | 75.12 |
| 0.1 | 2 | 0.7804 | 0.1968 | 74.78 |
| 0.1 | 3 | 0.7883 | 0.1970 | 75.01 |
| Mixup | 0 | / | 0.8125 | 0.8145 | -0.25 |
| 0.1 | 1 | 0.7932 | 0.2200 | 72.26 |
| 0.1 | 2 | 0.7869 | 0.1930 | 75.47 |
| 0.1 | 3 | 0.7886 | 0.1984 | 74.84 |

Now assume we have domain generalization model , training dataset and test dataset. Because backdoor implant only requiring training dataset which mixes with attack samples thought above conclusion, we can’t sure the model is clean, we should check whether the model is clean, in other word, we need to check whether the training dataset mixes with attack samples.

As previous works[ac作者,2018作者] show, the attack samples will be misclassified because they will provide a strong feature to force the model making wrong decision, and ac shows the strong feature can represent by using the activity value in the last hidden layer. Therefore, if we test domain one by one, and evaluate every sample’s activity value in each domain, we will conclude which domains are attacked. In this step, like ac had done, we use samples’ activity value, which all come from same domain and same class, to fit the cluster model, and we divide these sample into two groups. By using clustering result, we can evaluate which labels and domains are attacked. Because checking is one domain by one domain, the method can work as ac’s. What is different from ac is we use PCA to reduce dimensions not ICA, because our experiment shows PCA performs better than ICA. To evaluate the clustering’s result, we using silhouette score to evaluate whether the samples fit the clustering model well, furthermore, we can conclude whether the label and the domain are attacked. Silhouette score can be calculated in follow steps:

①:For each sample point , calculate the average distance between point and all other elements in the same group, denoted as ;

②:Select a group outside , calculate the average distance between and all points in , traverse all other groups, and find the nearest average distance, denoted as ;

③:For sample point , silhouette coefficient = ;

④:Calculate the silhouette coefficients of all , and the average value is the overall silhouette coefficient of the current groups, which measures the tightness of the data clustering

By setting experiment same above and using Mixup as method to generalize, we can solve question proposed above. The result[figure 1] shows by attacking different numbers of domain, there does exist abnormal silhouette score. This indicates that we can check whether the model is implanted backdoor and whether the training dataset mixes with attack samples.

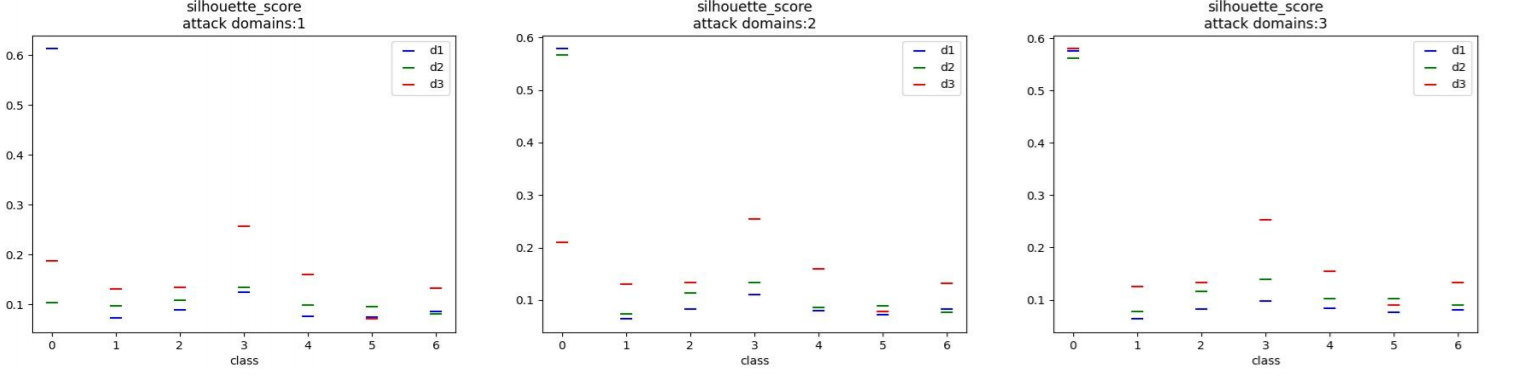


Figure 1: We plot the silhouette score of each domain and each label by using experiment setting above. We can easily observe there exist high value compared other values which are in low level when attack happens.

This method does have high accurate, but if some samples are marked wrong domain label, for example samples are from domain 0 but are marked into domain 1, will this cause abnormal silhouette. If the assumption is right, we don’t have enough assurance to conclude the model is implanted to backdoor and the dataset mixes with attack samples.

We collect clean samples which are same label to fit the clustering model, then dividing samples into two groups. We also use silhouette score to check whether the samples fit the model well. The result[table] indicates the samples which come from same label but different domain can’t fit the clustering model well, in other word, the difference between domain and domain doesn’t cause abnormal silhouette score. To visualize the difference between domain and domain, we take all the clean samples of a label, take out their activity value of last hidden layer, and select the two features that have been reduced dimension by PCA. We represent them on the coordinate axis. The result[table] also show the difference between domain and domain doesn’t cause abnormal silhouette score.

相同类的聚类结果（table）

相同类的聚类结果（figure）

Now, we can determine whether the model is implanted with backdoor, but to remove the attack samples or backdoor, we need to indicate which domains are attacked and which aren’t thought using silhouette score. Obviously, the abnormal silhouette score is usually high, but is there a possibility that samples, which are in same domain and label, with low silhouette score are mixed with a few attack samples. Put another way, what proportion of attack samples in dataset will cause this abnormal score.

The experiment setting is same as above. The number of samples used in cluster is 800, and they all come from same label. We change the proportion of attack samples, then divide them into two groups and calculate the silhouette score. The result[Table 2] show only samples are all clean, the score will be normal. This means using silhouette is enough to indicate which labels and domains are mixed with attack sample.

Table 2: The result of table is based on experiment setting above, we record the silhouette score from the proportion of attacked samples is 0 to 1, with gradually increasing 0.1 each step.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Proportion | 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| Score | 0.1155 | 0.5846 | 0.5778 | 0.5539 | 0.5479 | 0.5479 |
| Proportion | 0.6 | 0.7 | 0.8 | 0.9 | 1 |  |
| Score | 0.5546 | 0.5815 | 0.6261 | 0.6968 | 0.8028 |  |

# Algorithm

Detection and defense

In this section, we will detail the procession of backdoor defense. As showed above, samples of the same label but different domains don’t affect clustering, and we assume we have at least one domain sample that is not attacked.

We use all the data to train a model like normal training. Then we extract the activity values of all samples in the last hidden layer on the trained model, divide these activity values according to their domains and labels. For each domain and each label, we use clustering method to divide sample into two groups, calculate their silhouette score respectively and analysis whether these samples are mixed with attack samples. For each group, which is mixed with attack samples, we divide the samples into different groups by using multiple clustering with highest silhouette score. For each group, we mix it with samples which come from same label and clean domain, then we fit them in clustering model and calculate the silhouette. The groups with low silhouette score consist of clean sample. Until now, we can distinguish which samples are clean and remove attack samples. By using the clean samples, we can retrain model without backdoor.

In particularly, when dividing samples into different groups, we usually divide them into three or more groups. Because our experiment show dividing them into two groups can’t sperate clean samples and attack samples as much as possible. To achieve this goal, we divide samples into different numbers of groups and calculate their silhouette score respectively. Then we choose the number of groups with highest score as our clustering number. What’s more, our procession of removal uses the information from clean domain, which doesn’t need human’s interference compared to ac. This avoids subjectivity and improves detection’s efficiency.

# Experiment

In this section, we describe our experiment on PACS dataset, using a stand ResNet-18 model. The poisoned proportion is 10% and the target label is label 0. Meanwhile the training algorithms are ERM、Mixup、MMD and VREX.

.

Clustering number

Because the cluster number from ac[ac] can’t sperate clean samples from attack samples as much as possible. To try our best to find attack samples, we conduct experiment to find how many groups can separate attack samples from clean samples as much as possible.

We use the models produced by using ERM and Mixup, and just attack one domain. We collect the samples which are in attack domain and label, then divide them into 2~7 groups respectively and calculate their silhouette score.

The result[Table 3] shows dividing samples to three groups is well, and the score with dividing samples to two groups is lower than former. This may result from the attack method: the samples, whose origin label is equal to target label, will be attack, but the label has not changed in nature.

Table 3: The result is about the performance of clustering when we divide samples into different groups.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Cluster number(s) | 2 | 3 | 4 | 5 | 6 | 7 |
| Silhouette score |  |  |  |  |  |  |

Table1: Clustering number

Poisoned samples detect

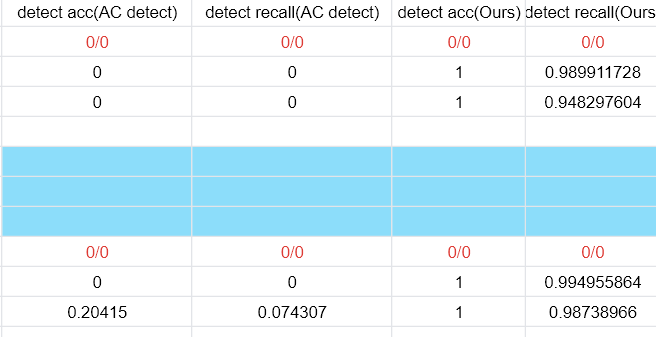
The procession of clustering only sperate clean samples and attack samples, so we need to identify groups which are clean and which are attacked.

This part’s experiment setting is same as experiment in cluster number, but we will show the detection performance of four methods of domain generalization.

Base on previous result in cluster number, the group with clean samples can be easily found by using our algorithm. The result[table] show only one group consists of most clean samples, that’s we can choose this group as our clean samples.

计算f1-score、accuracy precision 、recall

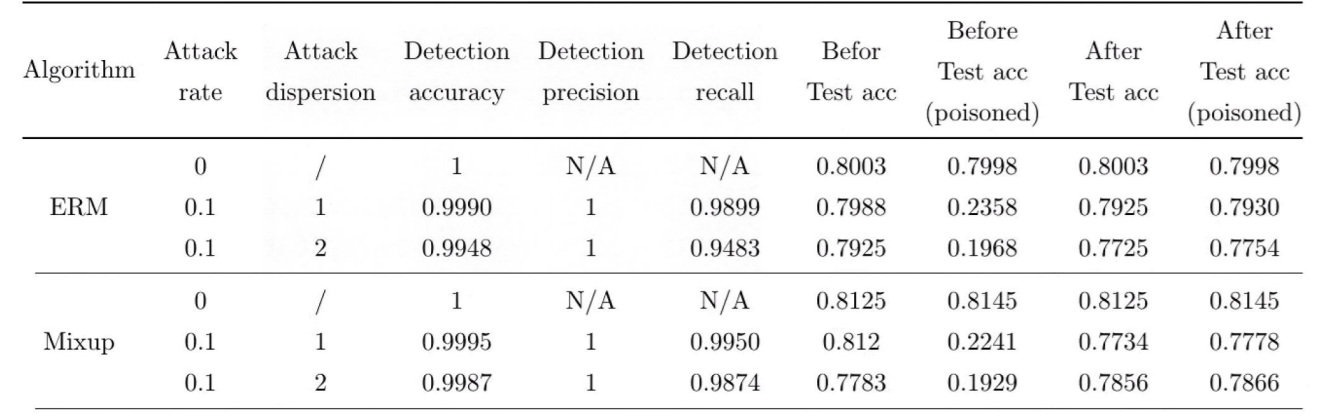
Ac和ours比较



#这里还缺一个table：干净数据和每一个group的聚类效果

Retrain model

After finding the attack sample, we remove it directly from the training set and retrain the model using the resulting training set without attack samples, and detect whether the model is implanted with a backdoor and the classification accuracy of clean samples. The extremely low backdoor trigger probability of the [table] shows that our method is very good at finding poisoned samples and removing the backdoor from the model.



1：Baracaldo, N., Chen, B., Ludwig, H., and Safavi, J. A. (2017).

Mitigating poisoning attacks on machine learning models:

A data provenance based approach. In Proceedings of the

10th ACM Workshop on Artificial Intelligence and Security,

pages 103–110. ACM.

2：1]Li Y, Jiang Y, Li Z, et al. Backdoor learning: A survey[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022.

123：G. Lovisotto, S. Eberz, and I. Martinovic, “Biometric Backdoors: A poisoning attack against unsupervised template updating,” arXiv preprint arXiv:1905.09162, 2019.

126：] Y. Kim, R. Daly, J. Kim, C. Fallin, J. H. Lee, D. Lee, C. Wilkerson, K. Lai, and O. Mutlu, “Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors,” ACM SIGARCH Computer Architecture News, vol. 42, no. 3, pp. 361–372, 2014

[19]Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-

pruning: Defending against backdooring attacks on deep

neural networks. In Research in Attacks, Intrusions, and

Defenses, pages 273–294. Springer International Publishing,

2018.

[20] Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma,

Y ousra Aafer, and Xiangyu Zhang. Abs: Scanning neu-

ral networks for back-doors by artificial brain stimulation.

In Proceedings of the ACM SIGSAC Conference on Com-

puter and Communications Security(CCS), page 1265–1282,

2019.

[21] Yingqi Liu, Shiqing Ma, Y ousra Aafer, Wen-Chuan Lee,

Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning

attack on neural networks. In Proceedings of the Annual Net-

work and Distributed System Security Symposium (NDSS),

2018.